REPORT DOCUMENTATION PAGE

FORM APPROVED
OMB NO. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 14, 2000	Final Report – 7/18/1995 – 7/17/1999

4. TITLE AND SUBTITLE
SCOPE – Scalable Computing Infrastructure
Tera-node Network Technology (Task 2)

5. FUNDING NUMBERS
DABT63-95-C-0095

6. AUTHORS

Neuman, B. Clifford, Gullapalli, Sridhar and Rao, Santosh

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

USC INFORMATION SCIENCES INSTITUTE
4676 ADMIRALTY WAY
MARINA DEL REY, CA 90292-6695

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Directorate of Contracting
Attention: ATZS-DKO-I
Post Office Box 12748
Fort Huachuca, Arizona 85670-2748

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12A. DISTRIBUTION/AVAILABILITY STATEMENT

UNCLASSIFIED/UNLIMITED

12B. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Scalable Computing Infrastructure project at the Information Sciences Institute of the University of Southern California investigated, developed, and deployed distributed systems software and services that enable the sharing of heterogeneous computing resources, within and across organizations, on an Internet-wide basis. These services enable cooperating organizations to establish agreements for helping one another in replacing computing capacity that is unavailable due to failure. It also enables individual users and organizations to purchase computing cycles from service providers to handle infrequent excess demand of applications when the frequency of such excess demands does not justify investment in permanent capacity. Software for SCOPE is was layered on top of, and includes extensions to, the Prospero Resource Manager (PRM). PRM was integrated with authentication and payment products and resource discovery services developed as part of related ISI projects.

14. SUBJECT TERMS
metacomputing, parallel computing, security, resource discovery, resource management, distributed systems

15. NUMBER OF PAGES
16

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

UNIVERSITY OF SOUTHERN CALIFORNIA *INFORMATION SCIENCES INSTITUTE*
4676 Admiralty Way, Marina del Rey, CA 90292

Tera-node Network Technology (TASK 2)
SCOPE - Scalable Computing Infrastructure
FINAL STATUS REPORT
PERIOD: JULY 18th, 1995 to JULY 17th, 1999

Sponsored by
Defense Advanced Research Projects Agency (DoD)
Information Technology Office
Issued by Directorate of ContractingFort Huachuca, AZ

Under Contract No.
DABT63-95-C-0095

Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Army Intelligence Center and Fort Huachuca Directorate of Contracting or the Defense Advanced Research Projects Agency.

Scalable Computing Infrastructure (SCOPE)
FINAL STATUS REPORT for DABT63-95-C-0095
Covering the period July 18th 1995 to July 17th 1999

Reported by: B. Clifford Neuman, Sridhar Gullapalli, and Santosh Rao

USC ISI Computer Networking Division

1. Introduction

Through the SCOPE project, ISI investigated, developed, and deployed distributed systems software and services that enable the sharing of large numbers of heterogeneous computing resources, within and across organizations, on an Internet-wide basis. These services enable cooperating organizations to establish mutual aid agreements for helping one another in replacing computing capacity that is unavailable due to failure. It also can enable individual users and organizations to purchase computing cycles from service providers to handle infrequent excess demand of applications when the frequency of such excess demands does not justify investment in permanent capacity. Software for SCOPE is was layered on top of, and includes extensions to, the Prospero Resource Manager (PRM). PRM was integrated with authentication and payment products and resource discovery services developed as part of related ISI projects.

2. Enhancements to PRM

In PRM, the functions of resource management are distributed across three types of managers: system managers, job managers, and node managers. One or more system managers control disjoint subsets of physical resources and are concerned with allocation of resources between jobs. Each job in the system is managed by a separate job manager, which is responsible for acquiring the resources needed by the job and assigning them to the individual tasks in the job. A node manager runs on each node in the system, managing resources local to the node and initiating tasks when requested by the job manager.

Parallel applications running under PRM are written in C and use the message-passing paradigm. The PRM software package includes communication libraries that provide the Message Passing Interface (MPI) and the Parallel Virtual Machine (PVM) communication interface. PRM communication is layered on top of the Asynchronous Reliable Delivery Protocol (ARDP), developed as part of related ISI projects.

As originally implemented PRM allowed users to execute applications on processors managed by the user's organization, or by a small set of organizations known by the user. The user's "virtual system" is configured to identify the system managers (not the individual computing resources) from which the job manager will request resources. This initial configuration did not scale across multiple administrative domains with differing administrative policies on the availability of computing resources. The computing infrastructure developed by SCOPE combined the elements of resource discovery to

identify resources capable of satisfying the needs of the application programs, resource allocation to acquire those resources once identified, authorization to restrict access to particular users or groups, and payment for the use of those resources.

3. SCOPE Innovations

- **Processor Cycles as a Commodity on the Internet:**

  The software infrastructure deployed by SCOPE enables computer owners to sell or barter processor cycles across the Internet. Individual users and organizations can purchase computer time on an as-needed basis. Organizations can set up cooperating environments that enable each to replace or supplement their computing capacities.

- **Computing Resource Directory Services:**

  A distributed directory service enables system managers to register the resources they manage, providing aggregate information about these resources, and the characteristics of the service itself, such as reliability guarantees and pricing. This service enables users to discover resources that can best satisfy their requirements for computing power, level of security, and cost of service.

- **Integration with Security Services:**

  The protocols for the computing infrastructure are being incorporated with the security services necessary to execute sensitive computations across open networks, select computing service providers that are trusted to protect the computation, and allow service providers to restrict access to particular users, all on a global scale.

- **Integration with Accounting Services:**

  The protocols for computing infrastructure are being integrated with multiple accounting and payment mechanisms to facilitate accounting and payment for use of a service provider's services and infrastructure.

4. Technology Transfer

Since the start of the SCOPE effort, the Globus system has evolved as the dominant platform for metacomputing applications. This has limited direct use of the Prospero Resource Manager, and we are no longer promoting PRM as a primary platform for use in metacomputing applications.

Recognizing the large community support for Globus, in the final phase of the SCOPE effort we began to transition the individual technologies developed as part of the SCOPE effort onto the Globus platform. This was made easier because ISI is one of the two principal players in the Globus system, and through contacts forged with he other players during our participation in the HPDC conference and program committees.

Our work on security and policy representation for the Prospero Resource Manager, using the GAA-API, has been integrated with the Globus system and is available in the current globus release. Since the SCOPE effort has concluded, additional development of the metacomputing related policies supported by the GAA-API is being conducted with funding from the Globus effort. Additional integration of components from the Scope effort with the Globus effort is anticipated.

5. Progress

YEAR ONE

During the initial year of the SCOPE effort, ISI prepared a preliminary release of the Propsero Resource Manager (PRM). Researchers worked on enhancements to PRM, including the addition of new features and services to improve security, configurability and ease of use:

- **Enhanced security features**

Configuration options were added to the node manager to enable PRM administrators to restrict the applications allowed to execute on the node. Depending on the level of security needed, the node manager can be configured at start-up to execute:

(a) only those programs whose executables resided in the PRM binaries directory,
(b) executables residing on the filesystem local to the site, or
(c) local executables as well as those downloaded from remote sites from which jobs are submitted.

A compile time configuration option which enables node and system managers to run under the userid root was developed. When configured in this manner, only those users with privileged access were able to start a PRM environment. This prevents unauthorized users from starting node and/or system managers on hosts that may have restrictions placed upon their use. PRM application processes are assigned one set of user identifiers specially created by the system administrator.

To prevent unauthorized requests from using computing resources, the System Manager was modified to use an access control list (ACL) to determine whether the host submitting a resource allocation request is authorized to use the resources. The ACL was initially specified in a file that the system manager reads at start-up. The mechanism provided is extensible, permitting access to resources for hosts in a particular domain.

- **Flexible configuration of PRM environment**

Support for dynamic addition and removal of node managers to/from the set of nodes was added to the system manager. To add itself to a resource pool, a node manager

contacts the system manager on the host named on the command line. A node manager may leave the pool or make itself temporarily unavailable during certain periods as configured at start-up.

- **Ease of use**

Users submitting PRM jobs are no longer required to start a node manager on their local host. The job manager checks for the existence of a local node manager. If no node manager is found, it takes on the responsibilities of spawning local tasks and translating task identifiers to host address-port pairs.

Several utility programs to assist the PRM administrator and end users were developed. These include prm_stat to query the status of a PRM system and report the availability of nodes, prm_jobs to obtain the status of one or more jobs belonging to a specific user, and prm_shutdown to gracefully shut down a PRM system.

- **Demo applications Brutessl were developed**

Brutessl is an externally developed program to do a brute force attack on 40-bit keys used in Netscape's Secure Sockets Layer (SSL) protocol. Its purpose is to demonstrate that export weakened cryptographic systems are not secure enough to be used for electronic commerce. The Brutessl program was adapted to run under PRM. This enabled a user to search several segments of the key space in parallel. As each instance of the program searches disjoint segments of the keyspace, linear speedup is achievable.

Povray is a public domain ray tracing program. A parallel version of Povray uses PVM library calls. Povray has been ported to the PRM environment and is being used as a test application to demonstrate the capabilities of PRM.

- **Release engineering and public release**

A subsequent version of the Prospero Resource Manager (PRM) was developed for public release. Efforts included fixing bugs, updating the documentation and release engineering to make the software package easy to install. Version 1.1 of PRM was released to the public in February, 1996. Source code, documentation and additional information were made available on the World Wide Web:

http://gost.isi.edu/gost-group/products/prm/prm-1.1.html

The February release also included patches and instructions for running the Brutessl software package on a set of machines in parallel using PRM. Brutessl is an externally developed program to do a brute force attack on 40-bit keys used in Netscape's Secure Sockets Layer (SSL) protocol.

- **Improved task management policies**

An enhanced resource allocation policy was designed and implemented to improve the utilization of computing resources and shorten the processing time. The Job Manager supports queueing of task creation requests when insufficient resources are available. Whenever a task is completed, and the computing node freed, the node is automatically reassigned a new task from the queue. This policy is especially useful for Master-Slave applications in which the Master assigns tasks from a task pool as, and when, resources become available.

- **New library functions**

A family of I/O functions was developed and implemented for handling standard I/O to tasks. These routines generalize the functions performed by the terminal I/O task, enabling any task to write to the standard input or read the standard output of any other task. These routines are useful for running applications that cannot be rewritten or relinked specifically for PRM.

- **PRM in use: Factoring the RSA-130 challenge**

ISI researchers collaborated with the FAFNER project in factoring the RSA-130 challenge continued during the reporting period. PRM was used to exploit idle processing resources at ISI and USC to solve factoring subproblems provided by a server at Cooperating Systems Corporation. Client machines (PRM nodes at ISI and USC) factored the numbers and returned results to the server. A heterogeneous collection of up to 14 Sparc workstations and six HP-700 workstations were utilized in this effort.

- **Web Interface to PRM**

A World Wide Web interface to the Prospero Resource Manager was developed, enabling users to start PRM jobs and view job output on a web browser. Job execution requests are submitted to the server through a forms interface. The web server executes a CGI script that starts up a job manager with the appropriate parameters. The script also sets up an I/O bridge between the PRM application and the web client.

For demonstration purposes, povray, a public domain ray tracing program was set up to execute under this configuration. The forms interface allows an image to be selected and submitted for tracing. A Java applet, downloaded from the web server, obtains the output of povray and displays it on the browser in real time.

A PRM status display page was added to assist the end-user in selecting the number of nodes and node architecture(s) on which to run his/her applications. The Java applet that displays rendered images was enhanced to provide real-time execution status in a separate popup window. This window displays the host names of nodes used in the computation, their architecture type and the percentage of the job completed by each node. Users have now been provided with the ability to upload their own input files (in povray input for-

mat) to the web server. A demonstration of this application was made available at the URL:

http://gost.isi.edu/gost-group/products/prm/prm-webdemo.html

- **Integration of PRM with new version of transport protocol**

ISI researchers integrated the then current internal release of PRM with Version 1 of the Asynchronous Reliable Delivery Protocol (ARDP Version 1). This version of ARDP provides a security context in the communication layer that serves as the building block for supporting mechanisms to encapsulate encryption, authentication, authorization, and payment information.

- **Deployment of PRM at USC**

ISI research began deployment PRM on campus workstations at USC. This deployment provided a testbed to evaluate PRM's scalability and use in real application environments. Towards this goal, ISI researchers met with staff from USC's University Computing Services to: i) identify applications that can benefit from PRM's resource management capabilities, ii) assess administrative support required for this deployment and continued operation, and, iii) determine policies for allocating resources to PRM applications.

YEAR TWO

During the second year of the SCOPE effort, ISI researchers continued development and extensions to the Prospero Resource Manager release.

- **Streamlined configuration procedure**

To simplify the software installation process, site dependent configuration parameters were eliminated from header files and placed in configuration scripts.

- **Dissertation completed and new staff brought on board**

Santosh Rao completed his dissertation and was awarded the Ph.D. degree from USC. He left the project on January 31, 1997 to work for Veritas on high performance file systems.

- **Improvements to the registration protocol:**

ISI researchers worked on and improved the registration protocol which enabled the node managers to link up with the system manager. Previously, at start-up time, the system manager would have to read the configuration file containing the host names of the machines running the node managers. In the new system, the node managers were able to register themselves at start-up with the system manager without requiring a special configuration file.

- **Improved the start-up procedure for node managers:**

ISI researchers developed interfaces to simplify launching of node managers. This included adding new options allowing the node manager to make the node available for running tasks only after the host machine has been idle for a user determinable period of time.

- **Developed additional test applications for PRM:**

ISI researchers developed a test application which uses a brute force approach to crack cryptographic keys used in the DES protocol. The keys we chose were 35-bits long and the key space was split into 2^13 segments of 2^22 keys each. A "master" program was devised to coordinate "worker programs" on nodes managed by PRM. Each worker program searches a particular segment exhaustively, and reports the result, be it failure or success, back to the master. A user can invoke the application by launching PRM's job manager and specifying a configuration file which contains the name of the "master" program and the number of nodes requested. PRM's system manager and node managers run the "worker" applications in parallel, transparent to the user, who only sees a speed-up in the computation. Fault tolerance of this application was improved by enabling restarting the search on a particular segment of the key space in case of task or machine failures.

- **Integrated the Prospero configuration API into PRM:**

ISI researched modified the configuration components of the Prospero Resource Manager to use the generic Prospero configuration API developed under another effort. This allowed users to create a configuration file called ".PRMResources" to set values for start up parameters for PRM. Command-line arguments were still supported and they override the values specified in the configuration file. New configuration options were added which enable users to specify limits on the resources consumed by PRM (e.g. CPU load, memory space). This was done after the initial deployment of PRM within the Networking Division at ISI, when it became apparent that the issue of resource consumption was very important to the owners of host machines.

- **Improvements to node allocation algorithm:**

ISI researchers changed the node allocation algorithm at the system manager, to prevent more than one node manager from running on a given host. In addition, in case a node manager on a particular host fails, the algorithm now enables the system manager's internal resource table entry for the host to be reused in the event of restarting a node manager on the host. This feature reduces the overhead involved in reallocating memory and other resources.

- **Design of ACL based security policy mechanism for use with PRM:**

This mechanisms allows users to restrict the types of applications they are willing to allow PRM to run on their hosts. The mechanism allows owners to specify restrictions for particular applications. Examples of such restrictions include:

- the users allowed to run the application.
- attributes of the application, e.g. version or architecture (SunOS, Solaris, HPUX etc.).
- identity of the endorser of the application, in the case of digitally signed certificates accompanying the application.
- the maximum CPU load the user is willing to provide to the PRM application.
- all the other options currently provided to the node manager as command line arguments or in a configuration file.

At start-up, a node manager reads a file containing a list of ACLs, one for each type of application the user is willing to run on the host machine. The location of the file will be known to the node manager via the Prospero Directory Service, and the file itself will be stored in the user's Prospero Virtual File System. Default configuration files will be created for each system manager, enabling easy sharing of the same security policy among multiple node managers. These files will be easily re-configurable by each user in order to suit particular security/usage requirements.

- **Authentication and verification issues for mobile code:**

In the second year, ISI researchers began to address issues related to authentication and verification of mobile code in heterogeneous computing environments. In the initial design, a principal to be authenticated is represented by a ("PROGRAM", <CHECKSUM>) tuple, where "PROGRAM" is a fixed string and <CHECKSUM> is an MD5 or similar checksum of the executable to be run by PRM. The access policy (implemented by an ACL) for a node will require that the executable be run on behalf of an authorized user and that the principal (PROGRAM, CHECKSUM) also be authorized to execute. The executable itself does not come with credentials certifying it as the tuple, but instead this credential is a proxy granted and signed by the program loader, allowing the executable to assume the identity (PROGRAM, CHECKSUM).

- **Integration of topology-d into PRM:**

Topology-d is a tool developed at the University of Southern California (USC) and enhanced by members of the GOST group. It estimates the state of networked resources by periodically computing the end-to-end latency and available bandwidth among them. Since these are end-to-end measurements, they capture information about both, the network and the server load. Based on these estimates, topology-d computes a fault tolerant, high bandwidth, low delay topology connecting participating sites. Topologies are periodically re-computed to take into account network and server load dynamics.

A tool like topology-d is beneficial to PRM because it offers the node managers a choice of a system manager based on the estimated performance of the network, in terms of bandwidth and latency. Topology-d's performance was tested during an intensive Internet-wide experiment and the results were satisfactory; topology-d has also been integrated into the Globus metacomputing system.

ISI researchers investigated different methods for integrating topology-d into PRM. One way was to publish topology-d's estimates into the Prospero Directory Service and have the node managers consult the Directory Service before choosing a system manager. A second method involved defining an Application Programming Interface (API) for topology-d, to be used by all Prospero applications, including PRM.

- **Participation in survey on computing frameworks**

ISI researchers were invited to participate in a survey of "computing frameworks" conducted by Mark Baker and Geoffrey Fox from NPAC, Syracuse University in August 1997. The purpose of the survey was to gather information for a research paper studying "computing frameworks", i.e. distributed environments which incorporate security, file sharing, resource management and scheduling as well as other tools and utilities. The results of the survey can be seen at:
http://www.sis.port.ac.uk/~mab/Survey/survey_results.html

YEAR THREE AND RESIDUAL EFFORT IN YEAR FOUR

During the third and final year of the SCOPE effort, ISI researchers continued development and extensions to the Prospero Resource Manager release.

- **Authentication and validation of mobile/downloadable executable content:**

An important issue considered by metacomputing environments like PRM is the authentication and verification of the code to be executed on the managed resources. The first hard problem considered here was how to strongly authenticate users and organizations before granting them access to the resources. After successful authentication, a strong authorization mechanism was needed to grant access to subsets of resources, depending on the identity of the authenticated users. We have developed a distributed authorization model that is currently being integrated with PRM.

Even after appropriate authentication and authorization, the system should be able to make sure that the code which is executed on the local resources will not be malicious. This is perhaps the hardest problem to solve. As a first step towards tackling it, we studied the existing literature in this area and put together a survey of techniques used by other researchers. Many of these techniques were not satisfactory, because they rely on kernel-specific mechanisms. This greatly limited their utility in a heterogeneous environment in which different operating systems might coexist. Information from this survey was made available at: http://gost.isi.edu/products/prm/security_survey.html

- **Implementation of Access Control List (ACL) architecture:**

The EACL (Extended Access Control List) framework designed by the GOST group in other related DARPA projects was used as the starting point for specifying access policies for use of nodes managed by the Prospero Resource Manager. This framework was modified to meet the specific requirements of PRM.

- **Discovery of Computing Resources**

In the initial version of PRM, there was a static relationship between the system manager, which is responsible for a set of computational resources, and the job manager, which requires resources. The host name of the system manager was usually set in a configuration file which was read when the job manager was launched. Massively parallel computations require more CPU power than can be provided by user-configured system managers. For this reason, a more dynamic b mechanism was needed that allowed job managers to discover additional resources when needed. This is the resource discovery problem addressed by PRM.

ISI researchers implemented a solution to this problem by using the Prospero Directory Service. Each system manager registers itself with the directory service. Besides the host name, the system manager can indicate various attributes of the set of resources it manages, such as number of processors, operating systems, platform characteristics. When a job manager needs more resources than its currently assigned system manager can offer, it will search the Prospero Directory Service for additional system managers. The query can be general, asking only for host names of available system managers, or it can be more specific, asking for system managers responsible for particular types of resources. For example, such a query might ask for all the system managers managing more than 256 nodes and running Sun Solaris as the operating system. After successful identification of appropriate system managers, the job is executed as usual by the node managers.

- **Bandwidth and Latency Estimation**

An issue related to resource discovery is server selection. Assuming the query mentioned above returns several system managers, the job manager should select the "best" one in terms of bandwidth and latency of the network link between itself and the system manager. One way to assess these values is for the job manager to periodically measure them and publish the results of the measurements in the directory service. We are working on the specification of a protocol for conducting and publishing these measurements. As a first step, we chose "netperf", a public-domain software package, as the bandwidth measurement tool used by PRM. Netperf is distributed as a stand-alone executable, but we transformed it into a library that is linked against PRM in order to avoid the overhead of launching a separate process. Initial experiments with the library have been very encouraging.

- **Distributed Authorization**

ISI researcher integrating a distributed authorization framework into PRM. The mechanism is based on two ideas:

i) Extended Access Control Lists (EACLs).

Conventional Access Control Lists (ACLs) were extended with an optional field added to each ACL entry specifying restrictions on authorized rights. In the case of PRM, the at-

tributes include strength of authentication, source of connection, limits on the physical resources managed by the system (e.g. CPU load, memory usage) and characteristics of applications that the users are willing to run on their processors (e.g. name, version, endorser).

ii) General Authorization and Access API (GAA API).

We have defined a common API to facilitate authorization decisions for applications. PRM invokes GAA API functions to determine if a requested operation or set of operation was authorized or if additional verification is necessary.

As an initial step towards integration, we developed a protocol for managing the EACL security policies. Since a security policy is often specified at the granularity of administrative domains, it made sense for the system manager to maintain a default EACL file which protects the resources within the domain. At the same time, it is possible for individual hosts and the node managers running on them to tailor the default policy to their specific needs. Our goal in designing a mechanism for the management of the EACL files was to enable easy sharing of a default policy among node managers, while allowing customization of the policy at the level of individual hosts.

EACL files are stored as objects within the Prospero Directory Service. A node ACL is represented by a link to a file, along with two attributes for the link. For example, before requesting resources from the node e.g. "syc.isi.edu", the system manager running on another node e.g. "darkstar.isi.edu" retrieves the EACL file associated with the node by looking for a link with attribute NODE_MANAGER. = syc.isi.edu. The EXTEND_DEFAULT link is used to decide whether the default or the local policy take precedence. If no such link is found, the default EACL file, provided for the domain will be used by retrieving the link with attribute SYSTEM_MANAGER = darkstar.isi.edu.

The following scenario shows how the management of the files is accomplished. For simplicity, we assume that the Prospero server is running on each host which runs the system manager.

a) The administrator of the domain whose resources are managed by a system manager running on host (called A) creates an EACL file describing the default authorization policy which applies to the domain.

b) The administrator registers with the Prospero server. A script takes as input the location of the EACL file and does the following:

- creates a Prospero object representing a link to the EACL file
- creates two attributes for the link: SYSTEM_MANAGER A EACL_DEFAULT True

c) If the administrator of a particular host B in the domain managed by A wishes to specify a local authorization policy different from the default one, a similar procedure is followed, except that the link to the local EACL file is created with the following attributes:

NODE_MANAGER B EXTEND_DEFAULT True/False

(True if the local policy extends the default one and False if the local policy completely replaces the default)

d) When a system manager receives a request for resources from a job manager, it calls a function to get the associated ACL. This function returns a handle to the EACL.

Sharing of security policies among node managers is made possible by associating a default EACL with each host. The default EACL is maintained by the domain administrator and lists domain-specific policies. Customization of the policies at the level of individual hosts is implemented through local EACLs, maintained by the host administrator.

To retrieve policies, one looks for a link with attribute NODE_MANAGER = B. If no such link is found, the default EACL file provided for the domain will be used. The file is retrieved by looking for a link with attributes SYSTEM_MANAGER = A and EACL_DEFAULT = True. If a link with NODE_MANAGER = B is found, then a second query is issued for the value of the attribute EXTEND_DEFAULT. If the value is True, the system manager will have to retrieve the default EACL file first, and then add the contents of the EACL file for node B. If the value is False, then only the EACL file for node B will be retrieved.

e) The principal's identity is placed into the security context by the transport protocol (ARDP) during authentication. The system manager calls a function to verify if the principal is allowed to run jobs. The security context and the EACL handle are passed as inputs to this function, which returns YES, NO or MAYBE (the latter is returned in the case when additional checks have to be made by the application).

f) After retrieval of the EACL file and the evaluation of creditions to determine the users identity, evaluation of the conditions listed in the file follows. If all the conditions are met, the job manager is allowed to use the resources on that particular host.

g) During the execution of tasks on a particular host, the node manager periodically checks whether the task is abiding to the limits imposed on the local resources. If it is not, then the task is interrupted and the job manager is notified.

Conditions implemented so far include the time window when the job is allowed to run, the minimum machine idle time before a job is allowed to run, and maximum CPU usage a job is allowed to take.

6. Trip Reports

During the course of the contract researchers attended, participated in, and presented research results at various conferences. This section lists the travel that was performed for these purposes.

Traveler: Santosh Rao Purpose:
Supercomputing '95 in San Diego, CA.
Date: Dec 5-8, 1995

This conference enabled Santosh Rao to interact with researchers in the high-performance computing field and learn about cluster-computing projects in academia such as the National Scalable Cluster Project (NSCP) and in the industry such as the Load Sharing Facility (LSF).

We learned about ARPA's interest in the CAVE effort and subsequently investigated the applicability of PRM in managing processing resources used in immersive virtual environments. We followed up on the Network Enabled Factoring project (FAFNER, headed by Syracuse University and Bellcore), and started developing a PRM application that will enable us to collaborate in their efforts to factor the RSA-130 challenge.
Participant: Clifford Neuman
Purpose: Program Committee Meeting of 7th IEEE International Symposium
on High Performance Distributed Computing (HPDC-7)
Dates: April 15 - 18, 1998 Location: Chicago, Il.

Dr. Neuman attended and participated in a program committee meeting for the 7th Symposium on High Performance Distributed Computing where he provided feedback on papers accepted for the symposium. In addition, he discussed and obtained feedback from others reviewers in attendance on a paper that he co-authored with others in the project, which was accepted by the Symposium.

Traveler: Clifford Neuman
Purpose: ARPA Software PI meeting, San Antonio Texas
Dates: 2/13/96 - 2/16/96

Discussed SCOPE effort with ARPA program managers and other principal investigators doing work in parallel and distributed computing. Brought the SCOPE effort to the attention of other researchers, including those involved with the FAFNER project.

Traveler: Clifford Neuman
Purpose: ARPA Network PI meeting, Charleston, South Carolina
Dates: 2/25/96 - 2/28/96

Dr. Neuman presented the status and goals of the SCOPE project to the program manager and other attendees.

Participant: Clifford Neuman
Purpose: DARPA ITO Principal Investigator (PI) Meeting, Dallas, Texas
Dates: 10/6/96 - 10/10/96

At the Dallas ITO PI meeting, Dr. Clifford Neuman represented the SCOPE effort, and participated in discussions regarding quality of service for distributed systems.

Participant: Clifford Neuman
Purpose: DARPA Quorum Principal Investigator (PI) Meeting, Dallas, Texas
Dates: 12/9/96 - 12/11/96

At the Quorum PI meeting, Dr. Clifford Neuman represented the SCOPE effort, and participated in discussions regarding quality of service for distributed systems.

Traveller: Clifford Neuman
Purpose: To attend Workshop on Building a Computational Grid, Argonne, IL
Dates: 9/7 -9/10/97

Dr. Neuman participated in a computation grid workshop held at Argonne National Labs where he presented and led a discussion on the topic of security for parallel computing infrastructure. Other discussions at the workshop involved applications for the infrastructure and other components of the infrastructure including resource management, communication, and networking.

Travellers: Clifford Neuman & Grig Gheorghiu
Purpose: Southern California Parallel Computing Systems Seminar, ISI, Marina del Rey, CA
Dates: 9/22/97

Dr. Neuman and Grig Gheorghiu attended the Southern California Parallel Computing Systems seminar held at ISI. This meeting provided an opportunity to interact with others working on parallel computing systems and infrastructure in Southern California and potential collaborative efforts were identified. One such effort that we are considering involves integration of the Prospero Resource manager with a Java compute engine run within web browsers that will enable any user to establish and register a computing node simply by visiting a web page.

Traveller: Grig Gheorghiu
Purpose: Attend 6th IEEE International Symposium on High Performance Distributed Computing (HPDC-6), Portland, OR
Dates: 8/5 - 8/8/97

Grig Gheorghiu attended the HPDC-6 symposium in Portland, Oregon.The interaction provided through attendance at the conference provided insight on other distributed computing and distributed resource management efforts in the research community. Discussions with attendants of the conference also allowed dissemination of information

regarding our own SCOPE efforts, specifically on the Prospero Resource Manager. New ideas were acquired for the development of PRM.

7. Scalable Computing Infrastructure Staff

B. Clifford Neuman	Senior Project Leader and Computer Scientist
Joseph Bannister	Associate Division Director
Katia Obraczka	Computer Scientist
Sridhar Gullapalli	Systems Programmer
Tatyana Ryutov	Graduate Research Assistant
Grig Gheorghiu	Graduate Research Assistant
Shih-Hao Liu	Graduate Research Assistant
Santosh Rao	Graduate Research Assistant
Yongho Song	Graduate Research Assistant
Rebecca Jordan	Project Support
Jeanine Yamazaki	Project Support

Appendix:

- **Publications**

The following publications describing results from the SCOPE effort were prepared and published with support from the SCOPE contract.

Santosh Rao, "Program Development Tools for Distributed Parallel Systems". Dissertation, December, 1996.

G. Gheorghiu, T. Ryutov, and B.C. Neuman, "Authorization for Metacomputing Applications". Seventh IEEE International Symposium on High Performance Distributed Computing (HPDC 7), Chicago, Illinois. July 1998.

Tatyana Ryutov, Grig Gheorghiu and Clifford Neuman, "An Authorization Framework for Metacomputing Applications". Cluster Computing 2(1999). Pages 165-175.

Nader Salehi, Katia Obrazcka, and B. Clifford Neuman, "The Performance of a Reliable, Request-Response Transport Protocol". Proceedings of the fourth IEEE Symposium on Computers and Communications, Egypt, July 1999.

Tatyana Ryutov, and Clifford Neuman, "Representation and Evaluation of Security policies for Distributed system Services". In Proceedings of the DARPA Information Survivability Conference & Exposition, January 2000. Hilton Head, South Carolina.